

Analysis of Sequential Decoding Complexity

Using the Berry-Esseen Inequality*

Po-Ning Chen[†] (*Senior Member, IEEE*), Yunghsiang S. Han[‡] (*Member, IEEE*),
 Carlos, R. P. Hartmann[§] (*Fellow, IEEE*) and Hong-Bin Wu[†]

[†]Department of Communications Engineering
 National Chiao Tung University, Hsin Chu
 Taiwan 30050, R.O.C.

[‡]Graduate Institute of Communication Engineering
 National Taipei University, Taipei
 Taiwan, R.O.C.

[§]Department of Electrical Engineering and Computer
 Science, Syracuse University, Syracuse, NY 13244-4100, USA

Subject Area: Coding Theory

Corresponding Author: Prof. Yunghsiang S. Han

Graduate Institute of Communication Engineering

National Taipei University

Taipei, Taiwan, R.O.C.

E-mail: yshan@mail.ntpu.edu.tw

Tel:+1-886-2-86746401 Fax:+1-886-2-26744448

Abstract

This study presents a novel technique to estimate the computational complexity of sequential decoding using the Berry-Esseen theorem. Unlike the theoretical bounds determined by the conventional central limit theorem argument, which often holds only for sufficiently large codeword length, the new bound obtained from the Berry-Esseen theorem is valid for any blocklength. The accuracy of the new bound is then examined for two sequential decoding algorithms, an ordering-free variant of the generalized Dijkstra's algorithm (GDA)(or simplified GDA) and the maximum-likelihood sequential decoding algorithm (MLSDA). Empirically investigating codes of small blocklength reveals that the theoretical upper bound for the simplified GDA almost matches the simulation results as the signal-to-noise ratio (SNR) per information bit (γ_b) is greater than or equal to 8 dB. However, the theoretical bound may become markedly higher than the simulated average complexity when γ_b is small. For the MLSDA, the theoretical upper bound is quite close to the simulation results for both high SNR ($\gamma_b \geq 6$ dB) and low SNR ($\gamma_b \leq 2$ dB). Even for moderate SNR, the simulation results and the theoretical bound differ by at most 0.8 on a \log_{10} scale.

Index Terms

Coding, Decoding, Large Deviations, Convolutional Codes, Maximum-Likelihood, Soft-Decision, Sequential Decoding

I. INTRODUCTION

The Berry-Esseen theorem [6, sec.XVI. 5] states that the distribution of the sum of independent zero-mean random variables $\{X_i\}_{i=1}^n$, normalized by the standard deviation of the sum, differs from the unit Gaussian distribution by no more than $C r_n / s_n^3$, where s_n^2 and r_n are, respectively, the sums of the marginal variances and the marginal absolute third moments, and the Berry-Esseen coefficient, C , is an absolute constant. Specifically, for every $a \in \mathfrak{R}$,

$$\left| \Pr \left\{ \frac{1}{s_n} (X_1 + \cdots + X_n) \leq a \right\} - \Phi(a) \right| \leq C \frac{r_n}{s_n^3}, \quad (1)$$

where $\Phi(\cdot)$ represents the unit Gaussian cumulative distribution function (cdf). The remarkable aspect of this theorem is that the upper bound depends only on the variance and the absolute third moment, and therefore, can provide a good probability estimate through the first three

*This work is supported by the *National Science Council* of Taiwan, R.O.C., under the projects of NSC 88-2219-E-009-004 and NSC 88-2213-E-260-006.

moments. A typical estimate of the absolute constant is six [6, sec.XVI. 5, Thm. 2]. When $\{X_n\}_{i=1}^n$ are identically distributed, in addition to independent, the absolute constant can be reduced to three, and has been reported to be improved down to 2.05 [6, sec.XVI. 5, Thm. 1]. In 1972, Beek sharpened the constant to 0.7975 [2]. Later, Shiganov further improved the constant down to 0.7915 for an independent sample sum, and, 0.7655, if these samples are also identically distributed [25]. Shiganov's result is generally considered to be the best result yet obtained thus far [24].

In applying this inequality to analyze the computational complexity of sequential decoding algorithms, the original analytical problem is first transformed into one that concerns the asymptotic probability mass of the sum of independent random samples. Inequality (1) can therefore be applied. The complexities of two sequential maximum-likelihood decoding algorithms are then analyzed. One is an ordering-free variant of the generalized Dijkstra's algorithm (GDA) [14] operated over a code tree of linear block codes, and the other is the maximum-likelihood sequential decoding algorithm (MLSDA) [13] that searches for the codeword over a trellis of binary convolutional codes.

The computational effort required by sequential decoding is conventionally determined using a random coding technique, which averages the computational effort over the ensemble of random tree codes [16], [18], [23]. Branching process analysis on sequential decoding complexity has been recently proposed [10], [19], [20]; the results, however, were still derived by averaging over semi-random tree codes. Chevillat and Costello proposed to analyze the computational effort of sequential decoding in terms of the column distance function of a specific time-invariant code [4]; but, the analysis only applied to a situation in which the code was transmitted via binary symmetric channels.

In light of the Berry-Esseen inequality and the large deviations technique, this work presents an alternative approach to derive the theoretical upper bounds on the computational effort of the simplified GDA and the MLSDA for binary codes antipodally transmitted through an additive white Gaussian noise (AWGN) channel. Unlike the bounds established in terms of the conventional central limit theorem argument, which often holds only for sufficiently large codeword length, the new bound is valid for any blocklength. Empirically investigating codes of small blocklength shows that for the trellis-based MLSDA, the theoretical upper bound is quite close to the simulation results for both high SNR and low SNR; even for moderate SNR, the

theoretical upper bound and the simulation results differ by no more than 0.579966 on a \log_{10} scale. For the tree-based ordering-free GDA, the theoretical bound coincides with the simulation results at high SNR; however, the bound tends to be substantially larger than the simulation results at very low SNR. The possible cause of the inaccuracy of the bound at low SNR for the tree-based ordering-free GDA is addressed at the end of this study.

The rest of this paper is organized as follows. Section II derives a probability bound for use of analyzing the sequential decoding complexity due to the Berry-Esseen inequality. Section III presents an analysis of the average computational complexity of the GDA. Section IV briefly introduces the MLSDA, and then analyzes its complexity upper bound. Conclusions are finally drawn in Section V.

Throughout this article, $\Phi(\cdot)$ denotes the unit Gaussian cdf.

II. BERRY-ESSEEN THEOREM AND PROBABILITY BOUND

This section derives an upper probability bound for the sum of independent random samples using the Berry-Esseen inequality. This bound is essential to the analysis of the computational effort of sequential decoding algorithms.

The approach used here is the *large deviations* technique, which is generally applied to compute the *exponent* of an exponentially decaying probability mass. The Berry-Esseen inequality is also applied to evaluate the *subexponential* detail of the concerned probability. With these two techniques, an upper bound of the concerned probability can be established.

Lemma 1: Let $Y_n = \sum_{i=1}^n X_i$ be the sum of i.i.d. random variables whose marginal distribution is $F(\cdot)$. Define the *twisted* distribution with parameter θ corresponding to $F(\cdot)$ as:

$$dF^{(\theta)}(x) \triangleq \frac{\exp\{\theta x\} dF(x)}{M(\theta)},$$

where $M(\theta) \triangleq E[e^{\theta X_1}]$. Let the random variable with probability distribution $F^{(\theta)}(\cdot)$ be $X^{(\theta)}$.

Then, for every $\theta < 0$,

$$\Pr\{Y_n \leq -n\alpha\} \leq A_n(\theta, \alpha) e^{\theta\alpha n} M^n(\theta),$$

where $A_n(\theta, \alpha) = \min\{B_n(\theta, \alpha), 1\}$,

$$B_n(\theta, \alpha) \triangleq \begin{cases} \frac{\sigma(\theta)}{\sqrt{2\pi n}[(\mu(\theta) + \alpha) - \theta\sigma^2(\theta)]} e^{-(\mu(\theta) + \alpha)^2 n / [2\sigma^2(\theta)]} + 2C \frac{\rho(\theta)}{\sigma^3(\theta)\sqrt{n}}, & \text{if } \alpha > \theta\sigma^2(\theta) - \mu(\theta); \\ e^{\theta[\sigma^2(\theta) - 2(\mu(\theta) + \alpha)]n/2} + 2C \frac{\rho(\theta)}{\sigma^3(\theta)\sqrt{n}}, & \text{otherwise,} \end{cases}$$

$$\mu(\theta) = E[X^{(\theta)}], \quad \sigma^2(\theta) = E[|X^{(\theta)} - \mu(\theta)|^2], \quad \rho(\theta) = E[|X^{(\theta)} - \mu(\theta)|^3]$$

and $C = 0.7655$.

Proof: Define $F_n^{(\theta)}(y) = \Pr[X_1^{(\theta)} + X_2^{(\theta)} + \cdots + X_n^{(\theta)} \leq y]$, and let the distribution of $[(X_1^{(\theta)} - \mu(\theta)) + \cdots + (X_n^{(\theta)} - \mu(\theta))]/[\sigma(\theta)\sqrt{n}]$ be $H_n(\cdot)$, where in the evaluation of the above two statistics, $\{X_i^{(\theta)}\}_{i=1}^n$ are assumed independent with common marginal distribution $F^{(\theta)}(\cdot)$. Then, by denoting $Y_n^{(\theta)} = X_1^{(\theta)} + X_2^{(\theta)} + \cdots + X_n^{(\theta)}$, we obtain:

$$\begin{aligned} \Pr(Y_n \leq -n\alpha) &= \int_{[x_1+\cdots+x_n \leq -n\alpha]} dF(x_1)dF(x_2)\cdots dF(x_n) \\ &= M^n(\theta) \int_{[x_1+\cdots+x_n \leq -n\alpha]} e^{-\theta(x_1+\cdots+x_n)} dF^{(\theta)}(x_1)dF^{(\theta)}(x_2)\cdots dF^{(\theta)}(x_n) \\ &= M^n(\theta) E \left[e^{-\theta(X_1^{(\theta)}+\cdots+X_n^{(\theta)})} \mathbf{1}\{X_1^{(\theta)} + \cdots + X_n^{(\theta)} \leq -n\alpha\} \right] \\ &= M^n(\theta) E \left[e^{-\theta Y_n^{(\theta)}} \mathbf{1}\{Y_n^{(\theta)} \leq -n\alpha\} \right] \\ &= M^n(\theta) \int_{-\infty}^{-n\alpha} e^{-\theta y} dF_n^{(\theta)}(y) \quad (y \rightarrow \sigma(\theta)\sqrt{n}y' + \mu(\theta)n) \\ &= M^n(\theta) \int_{-\infty}^{-(\mu(\theta)+\alpha)\sqrt{n}/\sigma(\theta)} e^{-\theta\sigma(\theta)\sqrt{n}y' - \theta\mu(\theta)n} dH_n(y') \quad (2) \\ &= e^{\theta\alpha n} M^n(\theta) \int_{-\infty}^{-(\mu(\theta)+\alpha)\sqrt{n}/\sigma(\theta)} e^{-\theta\sigma(\theta)\sqrt{n}[y' + (\mu(\theta)+\alpha)\sqrt{n}/\sigma(\theta)]} dH_n(y'), \quad (3) \end{aligned}$$

where $\mathbf{1}\{\cdot\}$ is the set indicator function, and (2) follows from $H_n(y) = F_n^{(\theta)}(\sigma(\theta)\sqrt{n}y + \mu(\theta)n)$.

Integrating by parts on (3) with $\lambda(dy) \triangleq -\theta\sigma(\theta)\sqrt{n} \exp\{-\theta\sigma(\theta)\sqrt{n}[y + (\mu(\theta)+\alpha)\sqrt{n}/\sigma(\theta)]\} dy$

defined over $(-\infty, -(\mu(\theta) + \alpha)\sqrt{n}/\sigma(\theta)]$, and then applying equation (1) yields

$$\int_{-\infty}^{-(\mu(\theta)+\alpha)\sqrt{n}/\sigma(\theta)} e^{-\theta\sigma(\theta)\sqrt{n}[y+(\mu(\theta)+\alpha)\sqrt{n}/\sigma(\theta)]} dH_n(y) \quad (4)$$

$$\begin{aligned} &= \int_{-\infty}^{-(\mu(\theta)+\alpha)\sqrt{n}/\sigma(\theta)} \left[H_n \left(-\frac{(\mu(\theta) + \alpha)\sqrt{n}}{\sigma(\theta)} \right) - H_n(y) \right] \lambda(dy) \\ &\leq \int_{-\infty}^{-(\mu(\theta)+\alpha)\sqrt{n}/\sigma(\theta)} \left[\Phi \left(-\frac{(\mu(\theta) + \alpha)\sqrt{n}}{\sigma(\theta)} \right) - \Phi(y) + 2C \frac{\rho(\theta)}{\sigma^3(\theta)\sqrt{n}} \right] \lambda(dy) \\ &= \int_{-\infty}^{-(\mu(\theta)+\alpha)\sqrt{n}/\sigma(\theta)} \left[\Phi \left(-\frac{(\mu(\theta) + \alpha)\sqrt{n}}{\sigma(\theta)} \right) - \Phi(y) \right] \lambda(dy) + 2C \frac{\rho(\theta)}{\sigma^3(\theta)\sqrt{n}} \\ &= \int_{-\infty}^{-(\mu(\theta)+\alpha)\sqrt{n}/\sigma(\theta)} e^{-\theta\sigma(\theta)\sqrt{n}[y+(\mu(\theta)+\alpha)\sqrt{n}/\sigma(\theta)]} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy + 2C \frac{\rho(\theta)}{\sigma^3(\theta)\sqrt{n}} \end{aligned} \quad (5)$$

$$\begin{aligned} &= e^{\theta^2\sigma^2(\theta)n/2} e^{-\theta(\mu(\theta)+\alpha)n} \Phi \left(\theta\sigma(\theta)\sqrt{n} - \frac{(\mu(\theta) + \alpha)\sqrt{n}}{\sigma(\theta)} \right) + 2C \frac{\rho(\theta)}{\sigma^3(\theta)\sqrt{n}} \\ &\leq \begin{cases} \frac{\sigma(\theta)}{\sqrt{2\pi n}[(\mu(\theta) + \alpha) - \theta\sigma^2(\theta)]} e^{-(\mu(\theta)+\alpha)^2 n/[2\sigma^2(\theta)]} + 2C \frac{\rho(\theta)}{\sigma^3(\theta)\sqrt{n}}, & \text{if } \alpha > \theta\sigma^2(\theta) - \mu(\theta); \\ e^{\theta^2\sigma^2(\theta)n/2} e^{-\theta(\mu(\theta)+\alpha)n} + 2C \frac{\rho(\theta)}{\sigma^3(\theta)\sqrt{n}}, & \text{otherwise,} \end{cases} \end{aligned} \quad (6)$$

where (5) holds by, again, applying integration by part, and (6) follows from

$$\Phi(-u) \leq \frac{1}{\sqrt{2\pi u}} e^{-u^2/2} \quad \text{and} \quad \Phi(u) \leq 1 \quad \text{for } u > 0.$$

It remains to show that

$$\int_{-\infty}^{-(\mu(\theta)+\alpha)\sqrt{n}/\sigma(\theta)} e^{-\theta\sigma(\theta)\sqrt{n}[y+(\mu(\theta)+\alpha)\sqrt{n}/\sigma(\theta)]} dH_n(y) \leq 1,$$

which be established by observing that

$$e^{\theta\alpha n} M^n(\theta) \int_{-\infty}^{-(\mu(\theta)+\alpha)\sqrt{n}/\sigma(\theta)} e^{-\theta\sigma(\theta)\sqrt{n}[y+(\mu(\theta)+\alpha)\sqrt{n}/\sigma(\theta)]} dH_n(y) = \Pr\{Y_n \leq -n\alpha\} \quad (7)$$

$$\begin{aligned} &= \Pr\{e^{\theta(Y_n+n\alpha)} \geq 1\} \\ &\leq E[e^{\theta(Y_n+n\alpha)}] \\ &= e^{\theta\alpha n} M^n(\theta). \end{aligned} \quad (8)$$

■

Some remarks are made following Lemma 1 as follows. First, the upper probability bound in Lemma 1 consists of two parts, the exponentially decaying $e^{\theta\alpha n} M^n(\theta)$ and the subexponentially bounded $A_n(\theta, \alpha)$. When $\alpha > \theta\sigma^2(\theta) - \mu(\theta)$ and $\alpha \neq -\mu(\theta)$,

$$B_n(\theta, \alpha) = \frac{\sigma(\theta)}{\sqrt{2\pi n}[(\mu(\theta) + \alpha) - \theta\sigma^2(\theta)]} e^{-(\mu(\theta)+\alpha)^2 n/[2\sigma^2(\theta)]} + 2C \frac{\rho(\theta)}{\sigma^3(\theta)\sqrt{n}} \approx 2C \frac{\rho(\theta)}{\sigma^3(\theta)\sqrt{n}}$$

since the first term decays exponentially fast, and $B_n(\theta, \alpha)$ reduces to the Berry-Esseen probability bound. However, when θ is taken to satisfy $\mu(\theta) = -\alpha$,

$$B_n(\theta, \alpha) = \frac{1}{\sqrt{2\pi n}|\theta|\sigma(\theta)} + 2C \frac{\rho(\theta)}{\sigma^3(\theta)\sqrt{n}},$$

and a larger bound (than the Berry-Esseen one) is resulted. In either case, $B_n(\theta, \alpha)$ vanishes exactly at the speed of $1/\sqrt{n}$. Secondly, when $A_n(\theta, \alpha) = 1$, the upper probability bound reduces to the simple Chernoff bound $e^{\theta\alpha n}M^n(\theta)$ for which a four-line proof from (7) to (8) is sufficient [8, Eq. (5.4.9)], and is always valid for every $\theta < 0$, regardless of whether $\alpha > \theta\sigma^2(\theta) - \mu(\theta)$ or not.

The independent samples $\{X_i\}_{i=1}^n$ with which our decoding problems are concerned actually consist of two i.i.d. sequences, one of which is Gaussian distributed and the other is non-Gaussian distributed. One way to bound the desired probability of $\Pr[\sum_{i=1}^n X_i \leq 0]$ is to directly use the Berry-Esseen inequality for independent but non-identical samples (which can be done following similar proof of Lemma 1). However, in order to manage a better bound, we will apply Lemma 1 only to those non-Gaussian i.i.d. samples, and manipulate the remaining Gaussian samples directly by way of their known probability densities in the below lemma (cf. The derivation in (9)).

Lemma 2: Let $Y_n = \sum_{i=1}^n X_i$ be the sum of independent random variables $\{X_i\}_{i=1}^n$, among which $\{X_i\}_{i=1}^d$ are identically Gaussian distributed with positive mean μ and non-zero variance σ^2 , and $\{X_i\}_{i=d+1}^n$ have common marginal distribution as $\min\{X_1, 0\}$. Let $\gamma \triangleq (1/2)(\mu^2/\sigma^2)$.

Then

$$\Pr\{Y_n \leq 0\} \leq \mathcal{B}(d, n-d, \gamma),$$

where

$$\mathcal{B}(d, n-d, \gamma) = \begin{cases} \Phi(-\sqrt{2\gamma n}), & \text{if } d = n; \\ \Phi\left(-\frac{(n-d)\hat{\mu}+d\sqrt{2\gamma}}{\sqrt{d}}\right) + \tilde{A}_{n-d}(\lambda) \\ \times \left[\Phi(-\lambda)e^{-\gamma}e^{\lambda^2/2} + \Phi(\sqrt{2\gamma})\right]^{n-d} \\ \times e^{d(-\gamma+\lambda^2/2)}\Phi\left(\frac{(n-d)\hat{\mu}+\lambda d}{\sqrt{d}}\right), & \text{if } 1 > \frac{d}{n} \geq 1 - \frac{\sqrt{4\pi\gamma}e^\gamma}{1+\sqrt{4\pi\gamma}e^\gamma\Phi(\sqrt{2\gamma})}; \\ 1, & \text{otherwise,} \end{cases}$$

$$\begin{aligned}
a &\triangleq -\hat{\mu} + (\sqrt{2\gamma} - \lambda)\tilde{\sigma}^2(\lambda) + \tilde{\mu}(\lambda), \\
\tilde{A}_{n-d}(\lambda) &\triangleq \min \left(\mathbf{1}\{a > 0\} \left[\frac{\tilde{\sigma}(\lambda)}{a\sqrt{2\pi(n-d)}} + 2C \frac{\tilde{\rho}(\lambda)}{\tilde{\sigma}^3(\lambda)\sqrt{n-d}} \right] + \mathbf{1}\{a \leq 0\}, 1 \right), \\
\hat{\mu} &\triangleq E[X_{d+1}] = -(1/\sqrt{2\pi})e^{-\gamma} + \sqrt{2\gamma}\Phi(-\sqrt{2\gamma}), \\
\tilde{\mu}(\lambda) &= -\frac{d}{n-d}\lambda, \\
\tilde{\sigma}^2(\lambda) &\triangleq -\frac{d}{n-d} - \frac{nd}{(n-d)^2}\lambda^2 + \frac{n}{n-d}\frac{1}{1 + \sqrt{2\pi}\lambda e^\gamma \Phi(\sqrt{2\gamma})}, \\
\tilde{\rho}(\lambda) &\triangleq \frac{n}{(n-d)} \frac{\lambda}{[1 + \sqrt{2\pi}\lambda e^\gamma \Phi(\sqrt{2\gamma})]} \left\{ 1 - \frac{d(n+d)}{(n-d)^2}\lambda^2 \right. \\
&\quad + 2 \left[\frac{n^2}{(n-d)^2}\lambda^2 + 2 \right] e^{-d(2n-d)\lambda^2/[2(n-d)^2]} \\
&\quad - \frac{d}{n-d} \left[\frac{n+d}{n-d}\lambda^2 + 3 \right] \sqrt{2\pi}\lambda e^\gamma \Phi(\sqrt{2\gamma}) \\
&\quad \left. - \frac{2n}{n-d} \left[\frac{n^2}{(n-d)^2}\lambda^2 + 3 \right] \sqrt{2\pi}\lambda e^{\lambda^2/2} \Phi\left(-\frac{n}{n-d}\lambda\right) \right\},
\end{aligned}$$

and λ is the unique solution (in $[0, \sqrt{2\gamma}]$) of

$$\lambda e^{(1/2)\lambda^2} \Phi(-\lambda) = \frac{1}{\sqrt{2\pi}} \left(1 - \frac{d}{n} \right) - \frac{d}{n} e^\gamma \Phi(\sqrt{2\gamma}) \lambda.$$

Proof: Only the bound for $d < n$ is proved since the case of $d = n$ can be easily substantiated.

Let

$$\tilde{\mu}(\theta) = \frac{E[X_{d+1}^{(\theta)}]}{\sigma}, \quad \tilde{\sigma}(\theta) = \frac{\text{Var}[X_{d+1}^{(\theta)}]}{\sigma^2}, \quad \text{and} \quad \tilde{\rho}(\theta) = \frac{E[|X_{d+1}^{(\theta)} - E[X_{d+1}^{(\theta)}]|^3]}{\sigma^3},$$

and let $\hat{\mu} = E[X_{d+1}]/\sigma$. By noting that $(\mu/\sigma) = \sqrt{2\gamma}$, and for any $\theta < 0$ satisfying that

$$a \triangleq -\hat{\mu} - \sigma\theta\tilde{\sigma}^2(\theta) + \tilde{\mu}(\theta) > 0,$$

$\Pr(Y_n \leq 0)$ can be bounded by

$$\Pr(Y_n \leq 0)$$

$$\begin{aligned}
&= \Pr\{X_1 + \cdots + X_d + X_{d+1} + \cdots + X_n \leq 0\} \\
&= \int_{-\infty}^{\infty} \Pr\{X_{d+1} + \cdots + X_n \leq -x\} \frac{1}{\sqrt{2\pi d\sigma^2}} e^{-\frac{(x-d\mu)^2}{2d\sigma^2}} dx, \quad (x \rightarrow \sigma x') \\
&= \int_{-\infty}^{\infty} \Pr\{X_{d+1} + \cdots + X_n \leq -\sigma x'\} \frac{1}{\sqrt{2\pi d}} e^{-\frac{(x'-d\sqrt{2\gamma})^2}{2d}} dx', \quad (x' \rightarrow (n-d)x'') \\
&= \int_{-\infty}^{\infty} \Pr\{X_{d+1} + \cdots + X_n \leq -\sigma(n-d)x''\} \frac{1}{\sqrt{2\pi d/(n-d)^2}} e^{-\frac{(x''-d\sqrt{2\gamma}/(n-d))^2}{2d/(n-d)^2}} dx'' \\
&= \int_{-\infty}^{\sigma\theta\tilde{\sigma}^2(\theta)-\tilde{\mu}(\theta)+a} \Pr\{X_{d+1} + \cdots + X_n \leq -\sigma(n-d)x''\} \frac{1}{\sqrt{2\pi d/(n-d)^2}} e^{-\frac{(x''-d\sqrt{2\gamma}/(n-d))^2}{2d/(n-d)^2}} dx'' \\
&+ \int_{\sigma\theta\tilde{\sigma}^2(\theta)-\tilde{\mu}(\theta)+a}^{\infty} \Pr\{X_{d+1} + \cdots + X_n \leq -\sigma(n-d)x''\} \frac{1}{\sqrt{2\pi d/(n-d)^2}} e^{-\frac{(x''-d\sqrt{2\gamma}/(n-d))^2}{2d/(n-d)^2}} dx'' \\
&\leq \int_{-\infty}^{\sigma\theta\tilde{\sigma}^2(\theta)-\tilde{\mu}(\theta)+a} \frac{1}{\sqrt{2\pi d/(n-d)^2}} e^{-\frac{(x''-d\sqrt{2\gamma}/(n-d))^2}{2d/(n-d)^2}} dx'' \\
&+ \int_{\sigma\theta\tilde{\sigma}^2(\theta)-\tilde{\mu}(\theta)+a}^{\infty} \min\left(\frac{\tilde{\sigma}(\theta)}{a\sqrt{2\pi(n-d)}} e^{-(\tilde{\mu}(\theta)+x'')^2(n-d)/[2\tilde{\sigma}^2(\theta)]} + 2C\frac{\tilde{\rho}(\theta)}{\tilde{\sigma}^3(\theta)\sqrt{n-d}}, 1\right) \\
&\times e^{\theta\sigma(n-d)x''} M^{n-d}(\theta) \frac{1}{\sqrt{2\pi d/(n-d)^2}} e^{-\frac{(x''-d\sqrt{2\gamma}/(n-d))^2}{2d/(n-d)^2}} dx'', \tag{9}
\end{aligned}$$

where $M(\theta) = E[e^{\theta X_{d+1}}]$, and the last inequality follows from Lemma 1. Observe that

$$e^{-(\tilde{\mu}(\theta)+x'')^2(n-d)/[2\tilde{\sigma}^2(\theta)]} \leq 1.$$

Thus,

$$\begin{aligned}
\Pr(Y_n \leq 0) &\leq \int_{-\infty}^{-\hat{\mu}} \frac{1}{\sqrt{2\pi d/(n-d)^2}} e^{-\frac{(x''-d\sqrt{2\gamma}/(n-d))^2}{2d/(n-d)^2}} dx'' \\
&+ \int_{-\hat{\mu}}^{\infty} \min\left(\frac{\tilde{\sigma}(\theta)}{a\sqrt{2\pi(n-d)}} + 2C\frac{\tilde{\rho}(\theta)}{\tilde{\sigma}^3(\theta)\sqrt{n-d}}, 1\right) \\
&\times e^{\theta\sigma(n-d)x''} M^{n-d}(\theta) \frac{1}{\sqrt{2\pi d/(n-d)^2}} e^{-\frac{(x''-d\sqrt{2\gamma}/(n-d))^2}{2d/(n-d)^2}} dx'' \\
&= \Phi\left(-\frac{(n-d)\hat{\mu} + d\sqrt{2\gamma}}{\sqrt{d}}\right) \\
&+ \tilde{A}_{n-d}(\theta) M^{n-d}(\theta) e^{d(\theta\sigma\sqrt{2\gamma} + \theta^2\sigma^2/2)} \Phi\left(\frac{(n-d)\hat{\mu} + d\sqrt{2\gamma}}{\sqrt{d}} + \theta\sigma\sqrt{d}\right), \tag{10}
\end{aligned}$$

where for $a > 0$,

$$\tilde{A}_{n-d}(\theta) = \min \left(\frac{\tilde{\sigma}(\theta)}{a\sqrt{2\pi(n-d)}} + 2C \frac{\tilde{\rho}(\theta)}{\tilde{\sigma}^3(\theta)\sqrt{n-d}}, 1 \right).$$

Now for $\theta < 0$ and $a \leq 0$, we can use Chernoff bound in (9) instead, in which case the derivation up to (10) similarly follows with $\tilde{A}_{n-d}(\theta) = 1$.

We then note that

$$M^{n-d}(\theta) e^{d(\theta\sigma\sqrt{2\gamma} + \theta^2\sigma^2/2)}$$

is exactly the moment generating function of $Y_n = \sum_{i=1}^n X_i$; hence, if $E[Y_n] = d\mu + (n-d)\sigma\hat{\mu} > 0$, then the solution θ of $\partial E[e^{\theta Y_n}]/\partial\theta = 0$ is definitely negative.

For notational convenience, we let $\lambda = (\mu/\sigma) + \sigma\theta = \sqrt{2\gamma} + \sigma\theta$, and yield that

$$M(\theta) = \Phi(-\lambda) e^{-\gamma} e^{\lambda^2/2} + \Phi(\sqrt{2\gamma}) \quad \text{and} \quad e^{\theta\sigma\sqrt{2\gamma} + \theta^2\sigma^2/2} = e^{-\gamma} e^{\lambda^2/2}.$$

Accordingly, the chosen $\lambda = \sqrt{2\gamma} + \sigma\theta$ should satisfy

$$\frac{\partial \left(\left[\Phi(-\lambda) e^{-\gamma} e^{\lambda^2/2} + \Phi(\sqrt{2\gamma}) \right]^{n-d} e^{d(-\gamma + \lambda^2/2)} \right)}{\partial \lambda} = 0,$$

or equivalently,

$$e^{(1/2)\lambda^2} \Phi(-\lambda) = \frac{1}{\sqrt{2\pi}\lambda} \left(1 - \frac{d}{n} \right) - \frac{d}{n} e^\gamma \Phi(\sqrt{2\gamma}). \quad (11)$$

As it turns out, the solution $\lambda = \lambda(\gamma)$ of the above equation depends only on γ . Now, by replacing $e^{(1/2)\lambda^2} \Phi(-\lambda)$ with $(1 - d/n)/(\sqrt{2\pi}\lambda) - (d/n)e^\gamma \Phi(\sqrt{2\gamma})$, we obtain

$$\begin{aligned} \tilde{\mu}(\lambda) &= \left. \frac{E[X_{d+1}^{(\theta)}]}{\sigma} \right|_{\theta=(\lambda-\sqrt{2\gamma})/\sigma} = -\frac{d}{n-d}\lambda \\ \tilde{\sigma}^2(\lambda) &\triangleq \left. \frac{\text{Var}[X_{d+1}^{(\theta)}]}{\sigma^2} \right|_{\theta=(\lambda-\sqrt{2\gamma})/\sigma} \\ &= -\frac{d}{n-d} - \frac{nd}{(n-d)^2}\lambda^2 + \frac{n}{n-d} \frac{1}{1 + \sqrt{2\pi}\lambda e^\gamma \Phi(\sqrt{2\gamma})}, \end{aligned}$$

and

$$\begin{aligned}
\tilde{\rho}(\lambda) &\triangleq \left. \frac{E \left[\left| X_{d+1}^{(\theta)} - \hat{\mu} \right|^3 \right]}{\sigma^3} \right|_{\theta=(\lambda-\sqrt{2\gamma})/\sigma} \\
&= \frac{n}{(n-d)} \frac{\lambda}{[1 + \sqrt{2\pi} \lambda e^\gamma \Phi(\sqrt{2\gamma})]} \left\{ 1 - \frac{d(n+d)}{(n-d)^2} \lambda^2 \right. \\
&\quad + 2 \left[\frac{n^2}{(n-d)^2} \lambda^2 + 2 \right] e^{-d(2n-d)\lambda^2/[2(n-d)^2]} \\
&\quad - \frac{d}{n-d} \left[\frac{n+d}{n-d} \lambda^2 + 3 \right] \sqrt{2\pi} \lambda e^\gamma \Phi(\sqrt{2\gamma}) \\
&\quad \left. - \frac{2n}{n-d} \left[\frac{n^2}{(n-d)^2} \lambda^2 + 3 \right] \sqrt{2\pi} \lambda e^{\lambda^2/2} \Phi \left(-\frac{n}{n-d} \lambda \right) \right\}
\end{aligned}$$

Hence, the previously obtained upper bound for $\Pr(Y_n \leq 0)$ can be reformulated as

$$\begin{aligned}
&\Phi \left(-\frac{(n-d)\hat{\mu} + d\sqrt{2\gamma}}{\sqrt{d}} \right) \\
&+ \tilde{A}_{n-d}(\lambda) \left[\Phi(-\lambda) e^{-\gamma} e^{\lambda^2/2} + \Phi(\sqrt{2\gamma}) \right]^{n-d} e^{d(-\gamma+\lambda^2/2)} \Phi \left(\frac{(n-d)\hat{\mu} + \lambda d}{\sqrt{d}} \right),
\end{aligned}$$

where

$$\tilde{A}_{n-d}(\lambda) = \min \left(\mathbf{1}\{a > 0\} \left[\frac{\tilde{\sigma}(\lambda)}{a\sqrt{2\pi(n-d)}} + 2C \frac{\tilde{\rho}(\lambda)}{\tilde{\sigma}^3(\lambda)\sqrt{n-d}} \right] + \mathbf{1}\{a \leq 0\}, 1 \right).$$

Finally, a simple derivation yields

$$\begin{aligned}
E[Y_n] &= dE[X_1] + (n-d)E[X_{d+1}] \\
&= \sigma \left(d\sqrt{2\gamma} + (n-d) \left[-(1/\sqrt{2\pi})e^{-\gamma} + \sqrt{2\gamma}\Phi(-\sqrt{2\gamma}) \right] \right),
\end{aligned}$$

and hence, the condition of $E[Y_n] > 0$ can be equivalently replaced by

$$\frac{d}{n} \geq 1 - \frac{\sqrt{4\pi\gamma}e^\gamma}{1 + \sqrt{4\pi\gamma}e^\gamma\Phi(\sqrt{2\gamma})}.$$

■

Again, if the simple Chernoff inequality is used instead in the derivation of (9), the bound remains of the same form in Lemma 2 except that $\tilde{A}_{n-d}(\lambda)$ is always equal to one.

Empirical evaluations of $\tilde{A}_{n-d}(\lambda)$ in Figs. 1 and 2 indicates that when the sample number $n \leq 50$, $\tilde{A}_{n-d}(\lambda)$ will be close to 1, and the subexponential analysis based on the Berry-Esseen inequality does not help improving the upper probability bound. However, for a slightly larger

n such as $n = 200$, a visible reduction in the probability bound can be obtained through the introduction of the Berry-Esseen inequality.

One of the main studied subjects in this paper is to examine whether the introduction of the subexponential analysis can help improving the complexity bound at practical code length. The observation from Figs. 1 and 2 does coincide with what we obtained in later applications. That is, some visible improvement in complexity bound can really be obtained for a little larger codeword length in the MLSDA (specifically, $N = 2(60 + 6)$ or $2(100 + 6)$). However, since the simulated codes are only of lengths 24 and 48, no improvement can be observed for the GDA algorithm.

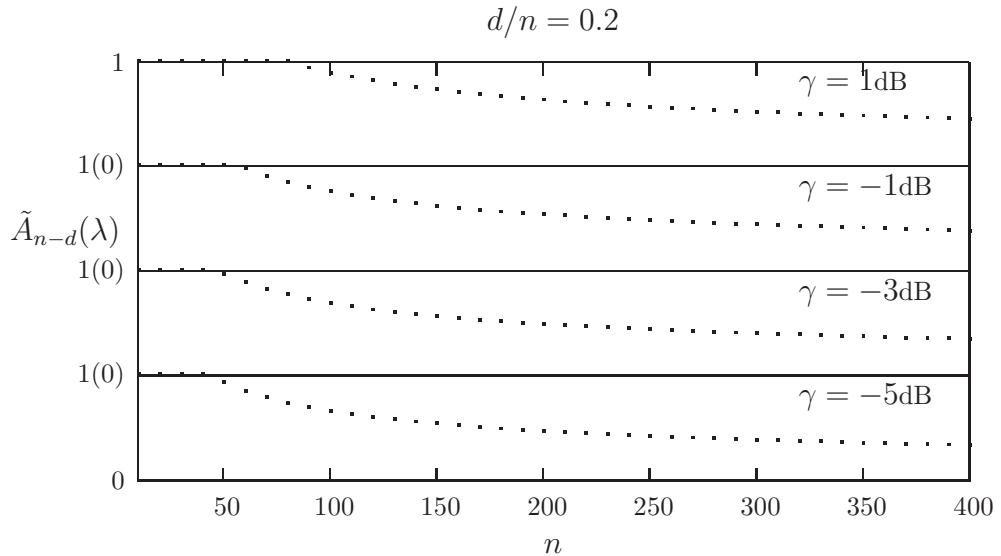


Fig. 1. $\tilde{A}_{n-d}(\lambda)$ for fixed $d/n = 0.2$ with respect to different γ . Notation “1(0)” represents that the y -tic is either 1 (for the curve below) or 0 (for the curve above).

We end this section by presenting the operational meanings of the three arguments in function $\mathcal{B}(\cdot, \cdot, \cdot)$ before their practice in subsequent sections. When in use for sequential-type decoding complexity analysis, the first integer argument is the Hamming distance between the transmitted codeword and the examined codeword up to the level of the currently visited tree node. The second integer argument represents a prediction of the future route, which is not yet occurred, and hence in our complexity analysis, is always equal to the maximum length of the codewords (resp. n for GDA algorithm and N for MLSDA algorithm) minus the length of the codeword

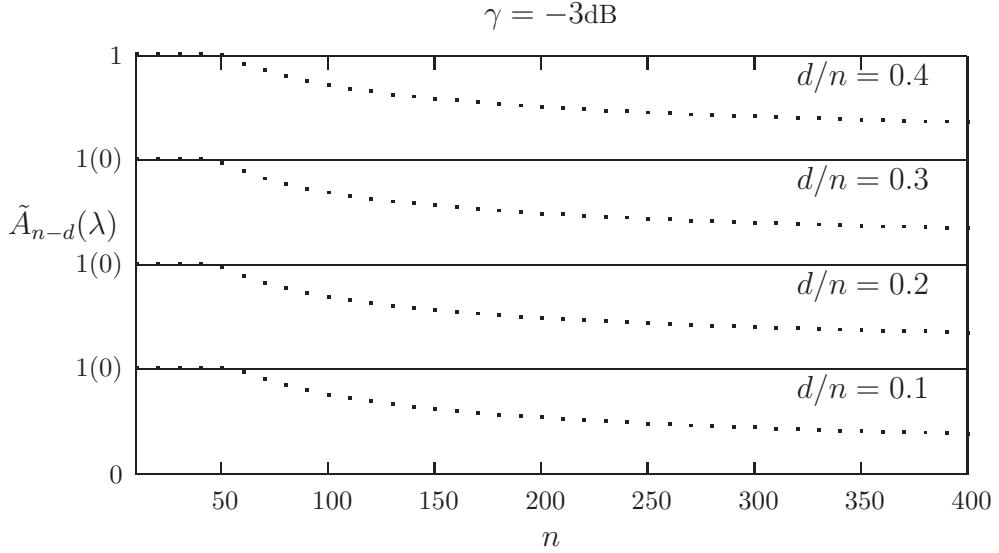


Fig. 2. $\tilde{A}_{n-d}(\lambda)$ for fixed $\gamma = -3dB$ with respect to different d/n ratios. Notation “1(0)” represents that the y -tic is either 1 (for the curve below) or 0 (for the curve above).

portion of the current visited node (resp. ℓ for GDA algorithm and ℓn for MLSDA algorithm).¹ The third argument is exactly the signal-to-noise ratio for the decoding environment, and is reasonably assumed to be always positive.

III. ANALYSIS OF THE COMPUTATIONAL EFFORT OF THE SIMPLIFIED GENERALIZED DIJKSTRA’S ALGORITHM

In 1993, a novel and fast maximum-likelihood soft-decision decoding algorithm for linear block codes was proposed in [14], and was called the generalized Dijkstra’s algorithm (GDA). Computer simulations have shown that the algorithm is highly efficient (that is, with small average computational effort) for certain number of linear block codes [5], [14]. Improvements of the GDA have been subsequently reported [1], [5], [9], [11], [15], [21], [26].

¹ The metric for use of sequential-type decoding can be generally divided into two parts, where the first part is determined by the *past* branches traversed thus far, while the second part helps predicting the *future* route to speed up the code search process [12]. For example, by adding a constant term $\sum_{i=1}^N \log_2 \Pr(y_i)$ to the accumulant Fano metric $\sum_{i=1}^q (\log_2[\Pr(y_j|b_j)/\Pr(y_j)] - R)$ up to level q , it can be seen that $\sum_{i=1}^q (\log_2[\Pr(y_j|b_j) - R])$ weights the history, and $\sum_{i=q+1}^N \log_2 \Pr(y_j)$ is the expectation of branch metrics to be added for possible future routes. Based on the intuition, the first argument and the second argument respectively realize the *historical* known part and the *future* predictive part of the decoding metric.

The authors of [15] proposed an upper bound on the average computational effort of an ordering-free variant of the GDA for linear block codes antipodally transmitted via the AWGN channel; however, the bound is valid only for codes with sufficiently large codeword length. In terms of the large deviations technique and Berry-Esseen inequality, an alternative upper bound that holds for *any* (thus including, *small*) codeword length can be given.

A. Notations and definitions

Let \mathcal{C} be an (n, k) binary linear block code with codeword length n and dimension k , and let $R \triangleq k/n$ be the *code rate* of \mathcal{C} . Denote the codeword of \mathcal{C} by $\mathbf{x} \triangleq (x_0, x_1, \dots, x_{n-1})$. Also, denote by $\mathbf{r} = (r_0, r_1, \dots, r_{n-1})$ the received vector due to a codeword of \mathcal{C} is transmitted via a time-discrete memoryless channel.

From [3] (also [27], [28]), the *maximum-likelihood* (ML) estimate $\hat{\mathbf{x}} = (\hat{x}_0, \hat{x}_1, \dots, \hat{x}_{n-1})$ for a time-discrete memoryless channel, upon the receipt of \mathbf{r} , satisfies

$$\sum_{j=0}^{n-1} (\phi_j - (-1)^{\hat{x}_j})^2 \leq \sum_{j=0}^{n-1} (\phi_j - (-1)^{x_j})^2 \quad \text{for all } \mathbf{x} \in \mathcal{C}, \quad (12)$$

where $\phi_j \triangleq \ln[\Pr(r_j|0)/\Pr(r_j|1)]$. An immediate implication of equation (12) is that using the log-likelihood ratio vector $\phi = (\phi_0, \phi_1, \dots, \phi_{n-1})$ rather than the received vector \mathbf{r} is sufficient in ML decoding.

When the linear block code is antipodally transmitted through the AWGN channel, the relationship between the binary codeword \mathbf{x} and the received vector \mathbf{r} can be characterized by

$$r_j = (-1)^{x_j} \sqrt{E} + e_j \quad \text{for } 0 \leq j \leq n-1, \quad (13)$$

where E is the signal energy per channel bit, and e_j represents a noise sample of a Gaussian process with single-sided noise power per hertz N_0 . The signal-to-noise ratio for the channel is therefore $\gamma \triangleq E/N_0$. In order to account for the code redundancy for different code rates, the SNR per information bit $\gamma_b = \gamma/R$ is used instead of γ in the following discussions.

A *code tree* of an (n, k) binary linear block code is formed by representing every codeword as a *code path* on a binary tree of $(n+1)$ levels. A *code path* is a particular *path* that begins at the *start node* at level 0, and ends at one of the *leaf nodes* at level n . There are two branches, respectively labelled by 0 and 1, that leave each node at the first k levels. The remaining nodes at levels k through $(n-1)$ consist of only a single leaving branch. The 2^k rightmost nodes

at level n are referred to as *goal nodes*. In notation, $\mathbf{x}_{[\ell]}$ is used to denote a path labelled by $(x_0, x_1, \dots, x_{\ell-1})$. For notational convenience, the subscript “[n]” is dropped for the label sequence of a code path, namely $\mathbf{x}_{[n]}$ is briefed by \mathbf{x} . The same notational convention is adopted for other notation including the received vector \mathbf{r} and the log-likelihood ratio vector ϕ .

B. Brief description of the GDA

For completeness, we brief the GDA decoding algorithm in [14] in this subsection.

After obtaining the log-likelihood ratio vector $\phi = (\phi_0, \phi_1, \dots, \phi_{n-1})$, the GDA algorithm first permutes the positions of codeword components such that the codeword component that corresponds to larger absolute value of log-likelihood ratio appears earlier in its position whenever possible, and still the first k positions uniquely determine a code path. The post-permutation codewords thereby result in a new code tree \mathcal{C}^* . Let $\phi^* \triangleq (\phi_0^*, \phi_1^*, \dots, \phi_{n-1}^*)$ be the new log-likelihood ratio vector after permutation, and define the path metric of a path $\mathbf{x}_{[\ell]}$ (over the new code tree \mathcal{C}^*) as $\sum_{j=0}^{\ell-1} (\phi_j^* - (-1)^{x_j})^2$. The path metric of a code path \mathbf{x} is thus given by $\sum_{j=0}^{n-1} (\phi_j^* - (-1)^{x_j})^2$. The algorithm then searches for the code path with the minimum path metric over \mathcal{C}^* , which, from equation (12), is exactly the code path labelled by the permuted ML codeword. As expected, the final step of the algorithm is to output the de-permuted version of the labels of the minimum-metric code path.

The search process of the GDA algorithm is guided by an evaluation function $f(\cdot)$, defined for all paths of a code tree. A simple evaluation function [11] that guarantees the ultimate finding of the minimum-metric code path is

$$f(\mathbf{x}_{[\ell]} | \phi^*) = \sum_{j=0}^{\ell-1} (\phi_j^* - (-1)^{x_j})^2 + \sum_{j=\ell}^{n-1} (|\phi_j^*| - 1)^2. \quad (14)$$

Hence, when a path $\mathbf{x}_{[\ell]}$ is extended to its immediate successor path $\mathbf{x}_{[\ell+1]}$, the evaluation function value is updated by adding the branch metric, $(\phi_\ell^* - (-1)^{x_\ell})^2 - (|\phi_\ell^*| - 1)^2$, to its original value. The algorithm begins the search from the path that contains only the start node. It then extends, among the paths that have been visited, the path with the smallest f -function value. Once the algorithm chooses to extend a path that ends at a goal node, the search process terminates. Notably, any path that ends at level k has already uniquely determined a code path. Hence, once a length- k path is visited and the f -function value associated with its respective code path does not exceed the associated f -function value of any of the later top paths in the stack, the

algorithm can ensure that this code path is the targeted one with the minimum code path metric. This indicates that the computational complexity of the GDA is dominantly contributed by those paths up to level k . This justifies our later analysis of the decoding complexity of the GDA, where only the computations due to those paths up to level k are considered.

The *simplified GDA algorithm* is an unpermuted variant of the GDA algorithm. In other words, its codeword search is operated over the unpermuted original code tree \mathcal{C} . Although both algorithms yield the same output, the simplified one was demonstrated to involve a larger branch metric computational load [15]. We quote the algorithm below.

- Step 1. Put the path that contains only the start node of the code tree into the Stack, and assign its evaluation function value as zero.*
- Step 2. Compute the evaluation function value (as in (14)) for each of the successor paths of the top path $x_{[\ell]}$ in the Stack by adding the branch metric of the extended branch to the evaluation function value of the top path. Delete the top path from the Stack.*
- Step 3. Insert the successor paths into the Stack in order of ascending evaluation function value.*
- Step 4. If the top path in the Stack ends at a goal node, output the codeword corresponding to the top path, and the algorithm stops; otherwise go to Step 2.*

It can be seen from the above algorithm that the simplified GDA algorithm resembles the stack algorithm except that it uses the evaluation function in (14) instead of the Fano metric to guide the search on the code tree, and is designed to decode the block codes rather than the convolutional codes. In addition, the simplified GDA algorithm is maximum-likelihood in performance as contrary to the sub-optimality of the stack algorithm.

C. Analysis of the computational effort of the simplified GDA

The computational effort of the simplified GDA can now be analyzed.

Theorem 1 (Complexity of the simplified GDA): Consider an (n, k) binary linear block code antipodally transmitted via an AWGN channel. The average number of branch metric computations evaluated by the simplified GDA, denoted by $L_{\text{SGDA}}(\gamma_b)$, is upper-bounded by

$$L_{\text{SGDA}}(\gamma_b) \leq 2 \sum_{\ell=0}^{k-1} \sum_{d=0}^{\ell} \binom{\ell}{d} \mathcal{B}(d, n - \ell, k\gamma_b/n), \quad (15)$$

where function $\mathcal{B}(\cdot, \cdot, \cdot)$ is defined in Lemma 2.

Proof: Assume without loss of generality that the all-zero codeword $\mathbf{0}$ is transmitted.

Let \mathbf{x}^* label the minimum-metric code path for a given log-likelihood ratio vector ϕ . Then we quote from [15] that for any path $\mathbf{x}_{[\ell]}$ selected for extension by the simplified GDA,

$$f(\mathbf{x}_{[\ell]}|\phi) \leq \sum_{j=0}^{n-1} (\phi_j - (-1)^{x_j^*})^2,$$

which implies that for $\ell < k$,

$$\begin{aligned} & \Pr [\text{path } \mathbf{x}_{[\ell]} \text{ is extended by the simplified GDA}] \\ & \leq \Pr \left[f(\mathbf{x}_{[\ell]}|\phi) \leq \sum_{j=0}^{n-1} (\phi_j - (-1)^{x_j^*})^2 \right] \end{aligned} \quad (16)$$

$$\leq \Pr \left[f(\mathbf{x}_{[\ell]}|\phi) \leq \sum_{j=0}^{n-1} (\phi_j - (-1)^0)^2 \right], \quad (17)$$

$$= \Pr \left[\sum_{j=0}^{\ell-1} (\phi_j - (-1)^{x_j})^2 + \sum_{j=\ell}^{n-1} (|\phi_j| - 1)^2 \leq \sum_{j=0}^{n-1} (\phi_j - 1)^2 \right], \quad (18)$$

where (17) follows from the assumption that the path metric of the \mathbf{x}^* -labelled code path is the smallest with respect to ϕ , and hence, does not exceed that of the 0-labelled code path.

Now denote by $\mathcal{J} = \mathcal{J}(\mathbf{x}_{[\ell]})$ the set of index j , where $0 \leq j \leq \ell - 1$, for which $x_j = 1$. Then (18) can be rewritten as

$$\begin{aligned} & \Pr [\text{path } \mathbf{x}_{[\ell]} \text{ is extended by the simplified GDA}] \\ & \leq \Pr \left[\sum_{j \in \mathcal{J}} \phi_j + \sum_{j=\ell}^{n-1} \min(\phi_j, 0) \leq 0 \right] \\ & = \Pr \left[\sum_{j \in \mathcal{J}} r_j + \sum_{j=\ell}^{n-1} \min(r_j, 0) \leq 0 \right] \end{aligned} \quad (19)$$

where (19) holds since for the AWGN channel specified in (13), $\phi_j = 4\sqrt{E}r_j/N_0$. As the all-zero codeword is assumed to be transmitted, r_j is Gaussian distributed with mean \sqrt{E} and variance $N_0/2$. Hence, Lemma 2 can be applied to obtain

$$\Pr [\text{path } \mathbf{x}_{[\ell]} \text{ is extended by the simplified GDA}] \leq \mathcal{B}(d, n - \ell, R\gamma_b),$$

where $d = |\mathcal{J}|$ is the Hamming weight of $\mathbf{x}_{[\ell]}$.

Observe that the extension of each path that ends at level ℓ , where $\ell < k$, causes two branch metric computations. Therefore, the expectation value of the number of branch metric evaluations

satisfies

$$L_{\text{SGDA}}(\gamma_b) \leq 2 \sum_{\ell=0}^{k-1} \sum_{d=0}^{\ell} \binom{\ell}{d} \mathcal{B}(d, n - \ell, R\gamma_b).$$

■

D. Numerical and simulation results

The accuracy of the previously derived theoretical upper bound for the average computational effort of the simplified GDA is now empirically studied. Two linear block codes are considered — one is a (24, 12) binary extended Golay code, and the other is a (48, 24) binary extended quadratic residue code.

Figures 3 and 4 illustrate the deviation between the simulated results and the theoretical upper bound in Theorem 1. Only one theoretical curve (rather than one enhanced by Berry-Esseen analysis and the other with simple Chernoff-based analysis) is plotted in the two figures because no improvement in function $\mathcal{B}(\cdot, \cdot, \cdot)$ can be obtained by the introduction of the Berry-Esseen analysis. According to these figures, the theoretical upper bound is quite close to the simulation results for high γ_b (above 8 dB). In such a case, the computational complexity of the simplified GDA reduces to its minimum possible values, 24 and 48, for (24, 12) and (48, 24) codes, respectively. As γ_b reaches 1 dB, the theoretical bound for (48, 24) code is around 12 times higher than the simulated average complexity. However, for the (24, 12) code, the theoretical bound and the simulation results differ only by 0.671638 on a \log_{10} scale at $\gamma_b = 1$ dB, and it is when $\gamma_b \leq -8$ dB that the upper bound becomes ten times larger than the simulated complexity. The conclusion section will address the possible cause of the inaccuracy of the theoretical bounds at low SNR.

IV. ANALYSIS OF THE COMPUTATIONAL EFFORT OF THE MLSDA

Based on the probability bound established in Lemma 2, the computational complexity of the maximum-likelihood sequential decoding algorithm (MLSDA) proposed in [13] is analyzed for convolutional codes antipodally transmitted via the AWGN channel.

A. Notation and definitions

Let \mathcal{C} be an (n, k, m) binary convolutional code, where k is the number of encoder inputs, n is the number of encoder outputs, and m is its *memory order* defined as the maximum number of

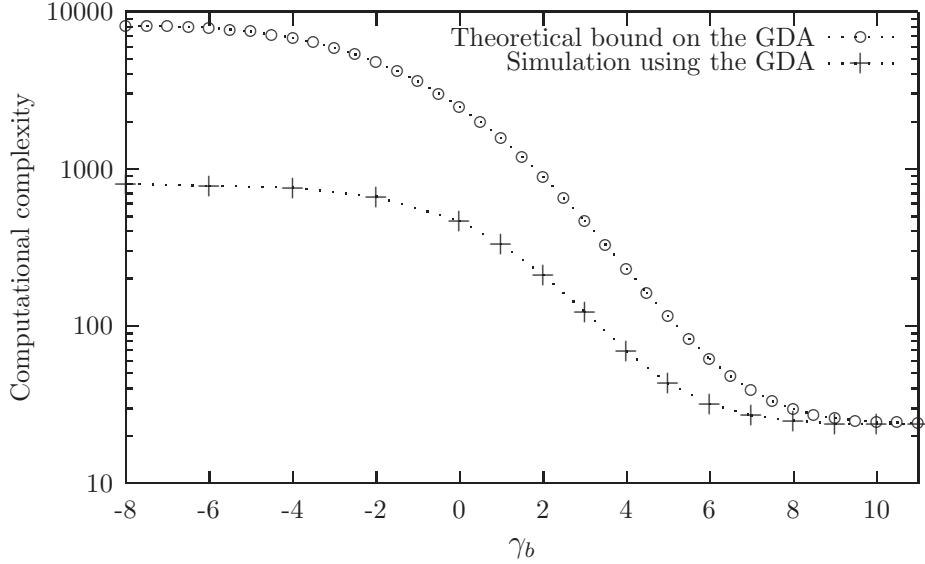


Fig. 3. Average computational complexity of the simplified GDA for (24, 12) binary extended Golay code.

shift register stages from an encoder input to an encoder output. Let $R \triangleq k/n$ and $N \triangleq n(L+m)$ be the *code rate* and the *code length* of \mathcal{C} , respectively, where L represents the length of applied information sequence. Denote the codeword of \mathcal{C} by $\mathbf{x} \triangleq (x_0, x_1, \dots, x_{N-1})$. Also denote the left portion of codeword \mathbf{x} by $\mathbf{x}_{(b)} \triangleq (x_0, x_1, \dots, x_b)$. Assume that antipodal signaling is used in the codeword transmission such that the relationship between binary channel codeword \mathbf{x} and received vector $\mathbf{r} \triangleq (r_0, r_1, \dots, r_{N-1})$ is

$$r_j = (-1)^{x_j} \sqrt{E} + e_j, \quad 0 \leq j \leq N-1, \quad (20)$$

where E is the signal energy per channel bit, and e_j is a noise sample of a Gaussian process with single-sided noise power per hertz N_0 . The signal-to-noise ratio per information bit $\gamma_b = (EN)/(N_0 kL)$ is again used to account for the code redundancy for various code rates.

A *trellis*, as depicted in Fig. 5 in terms of a specific example, can be obtained from a code tree by combining nodes with the same *state*. States are characterized by the content of the shift-register stages in a convolutional encoder. For convenience, the leftmost node (at level 0) and the rightmost node (at level $L+m$) of a trellis are named the *start node* and the *goal node*, respectively. A path on a trellis from the single start node to the single goal node is called a *code path*. Each branch in the trellis is labelled by an appropriate encoder output of length n .

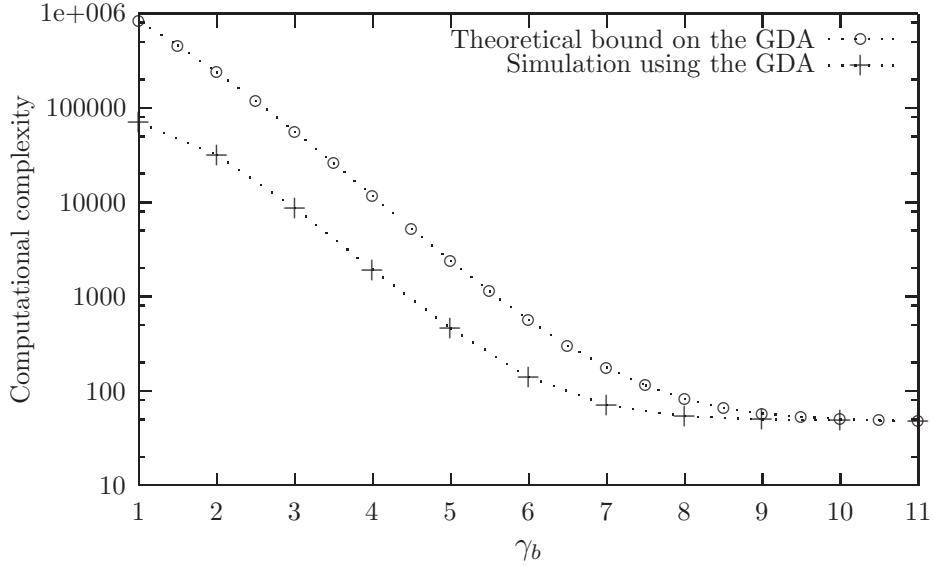


Fig. 4. Average computational complexity of the simplified GDA for (48, 24) binary extended quadratic residue code.

B. Maximum-likelihood soft-decision sequential decoding algorithm (MLSDA)

In [13], a trellis-based sequential decoding algorithm specifically for binary convolutional codes is proposed. The same paper proves that the algorithm performs maximum-likelihood decoding, and is thus named the *maximum-likelihood sequential decoding algorithm* (MLSDA). Unlike the conventional sequential decoding algorithm [7], [17], [22], [29] which requires only a single stack, the trellis-based MLSDA maintains two stacks — an *Open Stack* and a *Closed Stack*. For completeness, the algorithm is quoted below.

Step 1. Put the path that contains only the start node into the Open Stack, and assign its path metric as zero.

Step 2. Compute the path metric for each of the successor paths of the top path in the Open Stack by adding the branch metric of the extended branch to the path metric of the top path. Put into the Closed Stack both the state and level of the end node of the top path in the Open Stack. Delete the top path from the Open Stack.

Step 3. Discard any successor path that ends at a node that has the same state and level as any entry in the Closed Stack. If any successor path merges² with a path already in the Open

²“Merging” of two paths means that the two paths end at the same node.

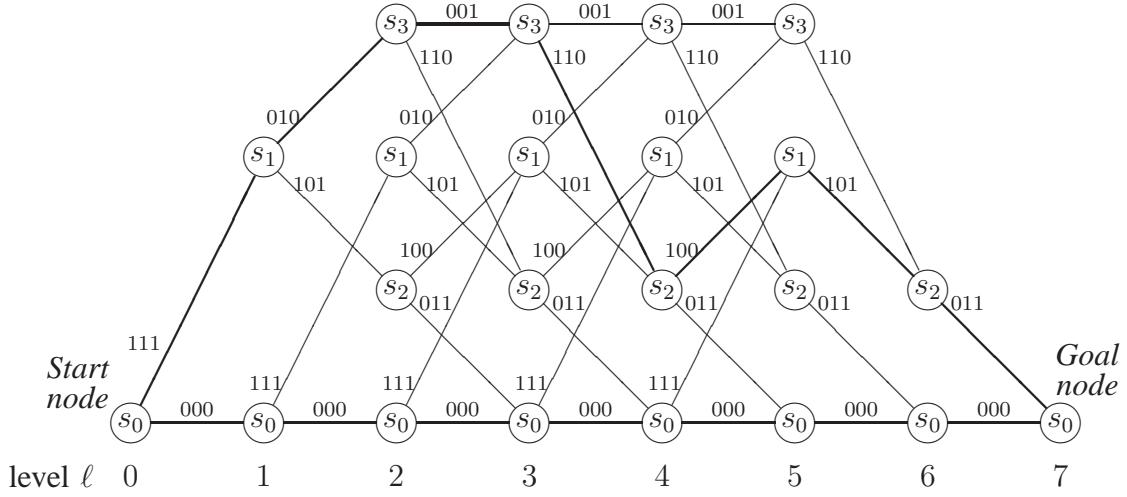


Fig. 5. Trellis for a $(3, 1, 2)$ binary convolutional code with information length $L = 5$. In this case, the code rate $R = 1/3$ and the codeword length $N = 3(5 + 2) = 21$. The code path indicated by the thick line is labelled by 111, 010, 001, 110, 100, 101 and 011, thus its corresponding codeword is $\mathbf{x} = (111010001110100101011)$.

Stack, eliminate the path with higher path metric.

Step 4. Insert the remaining successor paths into the Open Stack in order of ascending path metrics.

Step 5. If the top path in the Open Stack ends at the single goal node, the algorithm stops and output the codeword corresponding to the top path; otherwise go to Step 2.

We remark after the presentation of the MLSDA that the *Open Stack* contains all paths having been visited thus far, but excludes all prefixes of the paths in it. Hence, the *Open Stack* functions in a similar way as the stack in the conventional sequential decoding algorithm. The *Closed Stack* keeps the information of ending states and ending levels of those paths that had been the top paths of the *Open Stack* at some previous time. In addition, the path metric for a path labelled by $\mathbf{x}_{(\ell n-1)} = (x_0, x_1, \dots, x_{\ell n-1})$, upon receipt of $\phi_{(\ell n-1)}$, is given by

$$\zeta(\mathbf{x}_{(\ell n-1)} | \phi_{(\ell n-1)}) \triangleq \sum_{j=0}^{\ell n-1} (y_j \oplus x_j) \times |\phi_j|, \quad (21)$$

where $\phi_j = \log[\Pr(r_j|0)/\Pr(r_j|1)]$ is the j th received log-likelihood ratio, r_j is the j th received scalar, and $y_j = 1$ if $\phi_j < 0$ and $y_j = 0$, otherwise.

C. Analysis of the computational efforts of the MLSDA

Since the nodes at levels L through $(L+m-1)$ have only one branch leaving them, and L is typically much larger than m , the contribution of these nodes to the computational complexity due to path extensions can be reasonably neglected. Hence, the analysis in the following theorem only considers those branch metric computations applied up to level L of the trellis.

Notations that will be used in the next theorem are first introduced. Denote by $s_j(\ell)$ the node that is located at level ℓ and corresponds to state index j . Let $\mathcal{S}_j(\ell)$ be the set of paths that end at node $s_j(\ell)$. Also let $\mathcal{H}_j(\ell)$ be the set of the Hamming weights of the paths in $\mathcal{S}_j(\ell)$. Denote the minimum Hamming weight in $\mathcal{H}_j(\ell)$ by $d_j^*(\ell)$. As an example, $\mathcal{S}_3(3)$ equals $\{111010001, 000111010\}$ in Fig. 5, which results in $\mathcal{H}_3(3) = \{5, 4\}$ and $d_3^*(3) = 4$.

Theorem 2 (Complexity of the MLSDA): Consider an (n, k, m) binary convolutional code transmitted via an AWGN channel. The average number of branch metric computations evaluated by the MLSDA, denoted by $L_{\text{MLSDA}}(\gamma_b)$, is upper-bounded by

$$L_{\text{MLSDA}}(\gamma_b) \leq 2^k \sum_{\ell=0}^{L-1} \sum_{j=0}^{2^m-1} \mathcal{B} \left(d_j^*(\ell), N - \ell n, \frac{kL}{N} \gamma_b \right),$$

where if $\mathcal{H}_j(\ell)$ is empty, implying the non-existence of state j at level ℓ , then $\mathcal{B}(d_j^*(\ell), N - \ell n, kL\gamma_b/N) = 0$.

Proof: Assume without loss of generality that the all-zero codeword $\mathbf{0}$ is transmitted.

First, observe that for any two paths that end at a common node, only one of them will survive in the Open Stack. In other words, one of the two paths will be discarded either due to a larger path metric or because its end node has the same state and level as an entry in the Closed Stack. In the latter case, the surviving path has clearly reached the common end node earlier, and has already been extended by the MLSDA at some previous time (so that the state and level of its end node has already been stored in the Closed Stack). Accordingly, unlike the code tree search in the GDA, the branch metric computations that follow these two paths will only be performed once. It therefore suffices to derive the computational complexity of the MLSDA based on the nodes that have been extended rather than the paths that have been extended.

Let \mathbf{x}^* label the minimum-metric code path for a given log-likelihood ratio ϕ . Then we claim that if a node $s_j(\ell)$ is extended by the MLSDA, given that $\mathbf{x}_{(\ell n-1)}$ is the only surviving path (in

the Open Stack) that ends at this node at the time this node is extended, then

$$\zeta(\mathbf{x}_{(\ell n-1)}|\boldsymbol{\phi}_{(\ell n-1)}) \leq \zeta(\mathbf{x}^*|\boldsymbol{\phi}) \quad (22)$$

The validity of the above claim can be simply proved by contradiction. Suppose $\zeta(\mathbf{x}_{(\ell n-1)}|\boldsymbol{\phi}_{(\ell n-1)}) > \zeta(\mathbf{x}^*|\boldsymbol{\phi})$. Then the non-negativity of the individual metric $(y_j \oplus x_j)|\phi_j|$, which implies $\zeta(\mathbf{x}^*|\boldsymbol{\phi}) \geq \zeta(\mathbf{x}_{(b)}^*|\boldsymbol{\phi}_{(b)})$ for every $0 \leq b \leq N-1$, immediately gives $\zeta(\mathbf{x}_{(\ell n-1)}|\boldsymbol{\phi}_{(\ell n-1)}) > \zeta(\mathbf{x}_{(b)}^*|\boldsymbol{\phi}_{(b)})$ for every $0 \leq b \leq N-1$. Therefore, $\mathbf{x}_{(\ell n-1)}$ cannot be on top of the Open Stack (because some $\mathbf{x}_{(b)}^*$ always exists in the Open Stack), and hence violates the assumption that $s_j(\ell)$ is extended by the MLSDA.

For notational convenience, denote by $\mathcal{A}(s_j(\ell), \mathbf{x}_{(\ell n-1)})$ the *event* that “ $\mathbf{x}_{(\ell n-1)}$ is the only path in the intersection of $\mathcal{S}_j(\ell)$ and the Open Stack at the time node $s_j(\ell)$ is extended.” Notably,

$$\{\mathcal{A}(s_j(\ell), \mathbf{x}_{(\ell n-1)})\}_{\mathbf{x}_{(\ell n-1)} \in \mathcal{S}_j(\ell)}$$

are disjoint, and

$$\sum_{\mathbf{x}_{(\ell n-1)} \in \mathcal{S}_j(\ell)} \Pr \{ \mathcal{A}(s_j(\ell), \mathbf{x}_{(\ell n-1)}) \} = 1.$$

Then according to the above claim,

$$\begin{aligned} & \Pr \{ \text{node } s_j(\ell) \text{ is extended by the MLSDA} \} \\ &= \sum_{\mathbf{x}_{(\ell n-1)} \in \mathcal{S}_j(\ell)} \Pr \{ \mathcal{A}(s_j(\ell), \mathbf{x}_{(\ell n-1)}) \} \Pr \left\{ \begin{array}{l} \text{node } s_j(\ell) \text{ is extended} \\ \text{by the MLSDA} \end{array} \middle| \mathcal{A}(s_j(\ell), \mathbf{x}_{(\ell n-1)}) \right\} \\ &\leq \max_{\mathbf{x}_{(\ell n-1)} \in \mathcal{S}_j(\ell)} \Pr \left\{ \begin{array}{l} \text{node } s_j(\ell) \text{ is extended} \\ \text{by the MLSDA} \end{array} \middle| \mathcal{A}(s_j(\ell), \mathbf{x}_{(\ell n-1)}) \right\} \quad (23) \\ &\leq \max_{\mathbf{x}_{(\ell n-1)} \in \mathcal{S}_j(\ell)} \Pr \{ \zeta(\mathbf{x}_{(\ell n-1)}|\boldsymbol{\phi}_{(\ell n-1)}) \leq \zeta(\mathbf{x}^*|\boldsymbol{\phi}) \} \\ &\leq \max_{\mathbf{x}_{(\ell n-1)} \in \mathcal{S}_j(\ell)} \Pr \{ \zeta(\mathbf{x}_{(\ell n-1)}|\boldsymbol{\phi}_{(\ell n-1)}) \leq \zeta(\mathbf{0}|\boldsymbol{\phi}) \} \\ &= \max_{\mathbf{x}_{(\ell n-1)} \in \mathcal{S}_j(\ell)} \Pr \left\{ \sum_{j=0}^{\ell n-1} (y_j \oplus x_j)|\phi_j| \leq \sum_{j=0}^{N-1} (y_j \oplus 0)|\phi_j| \right\}, \end{aligned}$$

where the replacement of \mathbf{x}^* by the all-zero codeword $\mathbf{0}$ follows from $\zeta(\mathbf{x}^*|\boldsymbol{\phi}) \leq \zeta(\mathbf{0}|\boldsymbol{\phi})$. We then observe that for the AWGN channel defined through (20), $\phi_j = 4\sqrt{E}r_j/N_0$; hence, y_j can be determined by

$$y_j = \begin{cases} 1, & \text{if } r_j < 0; \\ 0, & \text{otherwise.} \end{cases}$$

This observation, together with the fact that $2(y_j \oplus x_j)|r_j| = r_j[(-1)^{y_j} - (-1)^{x_j}]$, gives

$$\begin{aligned}
& \Pr \{ \text{node } s_j(\ell) \text{ is extended by the MLSDA} \} \\
& \leq \max_{\mathbf{x}_{(\ell n-1)} \in \mathcal{S}_j(\ell)} \Pr \left\{ \sum_{j=0}^{\ell n-1} (y_j \oplus x_j)|r_j| \leq \sum_{j=0}^{N-1} (y_j \oplus 0)|r_j| \right\}, \\
& = \max_{\mathbf{x}_{(\ell n-1)} \in \mathcal{S}_j(\ell)} \Pr \left\{ \sum_{j=0}^{\ell n-1} r_j [(-1)^{y_j} - (-1)^{x_j}] \leq \sum_{j=0}^{N-1} r_j [(-1)^{y_j} - (-1)^0] \right\} \\
& = \max_{\mathbf{x}_{(\ell n-1)} \in \mathcal{S}_j(\ell)} \Pr \left\{ \sum_{j \in \mathcal{J}(\mathbf{x}_{(\ell n-1)})} r_j + \sum_{j=\ell n}^{N-1} \min(r_j, 0) \leq 0 \right\},
\end{aligned}$$

where $\mathcal{J}(\mathbf{x}_{(\ell n-1)})$ is the set of index j , where $0 \leq j \leq \ell n - 1$, for which $x_j = 1$. As r_j is Gaussian distributed with mean \sqrt{E} and variance $N_0/2$ due to the transmission of the all-zero codeword, Proposition 1 (in the Appendix) and Lemma 2 can be applied to obtain

$$\begin{aligned}
& \Pr \{ \text{node } s_j(\ell) \text{ is extended by the MLSDA} \} \\
& \leq \max_{d \in \mathcal{H}_j(\ell)} \Pr \left\{ r_1 + \cdots + r_d + \sum_{j=\ell n}^{N-1} \min(r_j, 0) \leq 0 \right\} \\
& = \Pr \left\{ r_1 + \cdots + r_{d_j^*(\ell)} + \sum_{j=\ell n}^{N-1} \min(r_j, 0) \leq 0 \right\} \\
& \leq \mathcal{B} \left(d_j^*(\ell), N - \ell n, \frac{kL}{N} \gamma_b \right).
\end{aligned}$$

Consequently,

$$L_{\text{MLSDA}}(\gamma_b) \leq 2^k \sum_{\ell=0}^{L-1} \sum_{j=0}^{2^m-1} \mathcal{B} \left(d_j^*(\ell), N - \ell n, \frac{kL}{N} \gamma_b \right),$$

where the multiplication of 2^k is due to the fact that whenever a node is extended, 2^k branch metric computations will follow. \blacksquare

D. Numerical and simulation results

The accuracy of the previously derived theoretical upper bound for the computational effort of the MLSDA is now empirically examined using two types of convolutional codes. One is a $(2, 1, 6)$ code with generators 634, 564 (octal); the other is a $(2, 1, 16)$ code with generators 1632044, 1145734 (octal). The lengths of the applied information bits are 60 and 100.

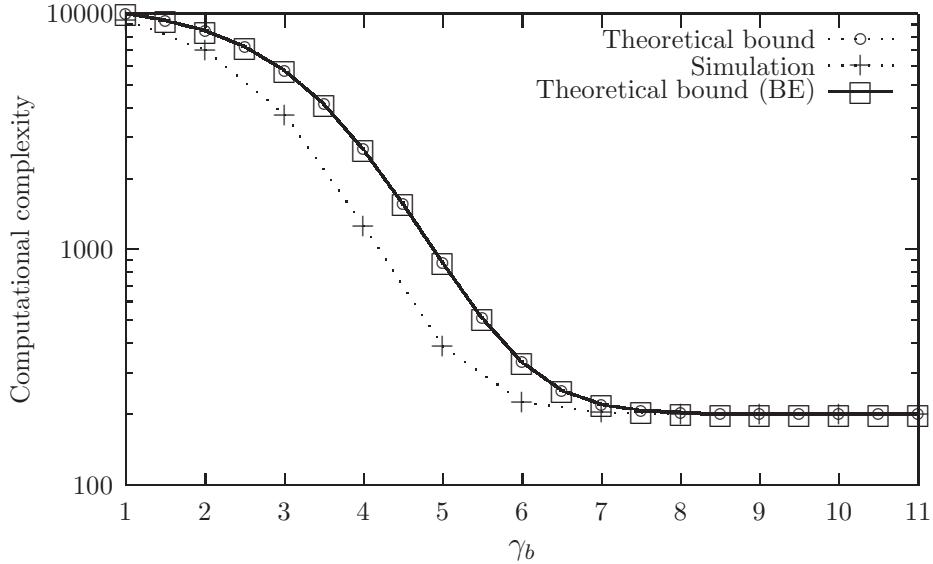


Fig. 6. Average computational complexity of the MLSDA for $(2, 1, 6)$ convolutional code with generators 634, 564 (octal) and information length $L = 100$.

Figures 6–9 present the deviation between the simulated results and the two theoretical upper bounds on the computational complexity of the MLSDA. According to these figures, the Berry-Esseen-enhanced theoretical upper bound is fairly close to the simulation results for both high γ_b (above 6 dB) and low γ_b (below 2 dB). Even for moderate γ_b , they only differ by no more than 0.8 for Figs. 6–9 on a \log_{10} scale. The differences between the two theoretical upper bounds with and without Berry-Esseen analysis are now visible in these figures. For example, the ratios of the two theoretical bounds are respectively 0.86, 0.90 and 0.95 at 4.0 dB, 4.5 dB and 5.0 dB in Fig. 8.

A side observation from these figures is that the codes with longer constraint length, although having a lower bit error rate, require more computations. However, such a tradeoff on constraint length and bit error rate can be moderately eased at high SNR. Notably, when $\gamma_b > 6$ dB, the average computational effort of the MLSDA in all four figures is reduced to approximately $2^k L$ in spite of the constraint length.

V. CONCLUSIONS

In terms of the large deviations technique and Berry-Esseen theorem, this study established theoretical upper bounds on the computational effort of the simplified GDA and the MLSDA

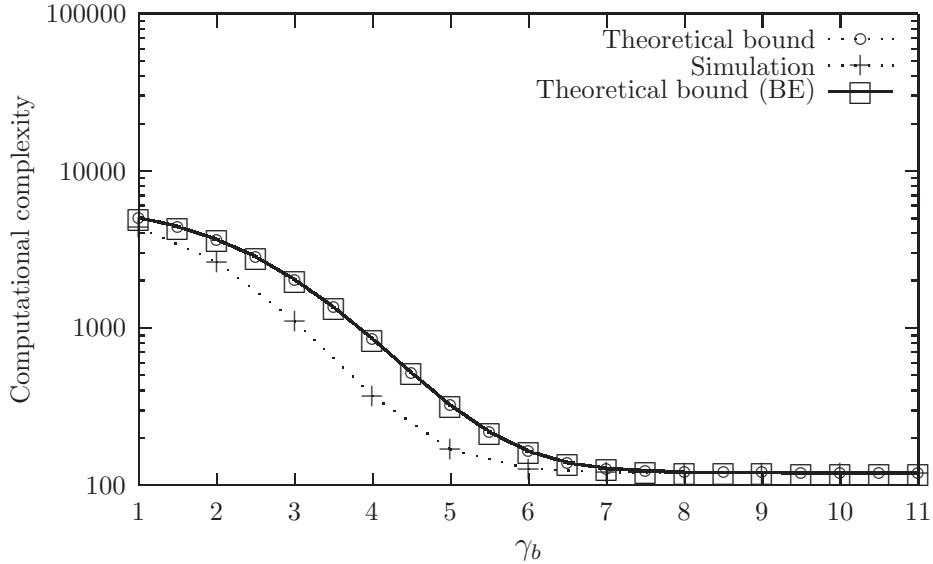


Fig. 7. Average computational complexity of the MLSDA for $(2, 1, 6)$ convolutional code with generators 634, 564 (octal) and information length $L = 60$.

for AWGN channels.

There may be two factors determining the accuracy of the complexity upper bound. The first factor is the accuracy of the large deviations probability bound for sum of independent samples in Lemma 2, and the second one is the accuracy of the estimate of the node extension probability for sequential-type decoding. We however found that the main inaccuracy may not come from the latter. Taking the GDA algorithm as an example, (16) is actually the exact event for path $x_{[\ell]}$ to be extended by the simplified GDA, and (17) becomes equality when the maximum-likelihood decision is exactly the transmitted all-zero codewords. Notably, as long as the node expanding distribution for each node is known, the average decoding complexity can be exactly obtained (specifically, if $Z_j = 1$ when node j is visited and expanded, and $Z_j = 0$, otherwise, then the average number of computations is exactly $2 \sum_j E[Z_j] = 2 \sum_j \Pr[Z_j = 1]$ since the extension of each path causes two branch metric computations). Hence, the main inaccuracy is due to the overestimate of the large deviations probability bound for sum of independent variables (and, of course, accumulating such overestimate by summing for all nodes may make worse the situation). Since the codes simulated for the GDA algorithm are of lengths 24 and 48 under which the large deviations probability bound is very inaccurate, the resultant complexity bound is also inaccurate,

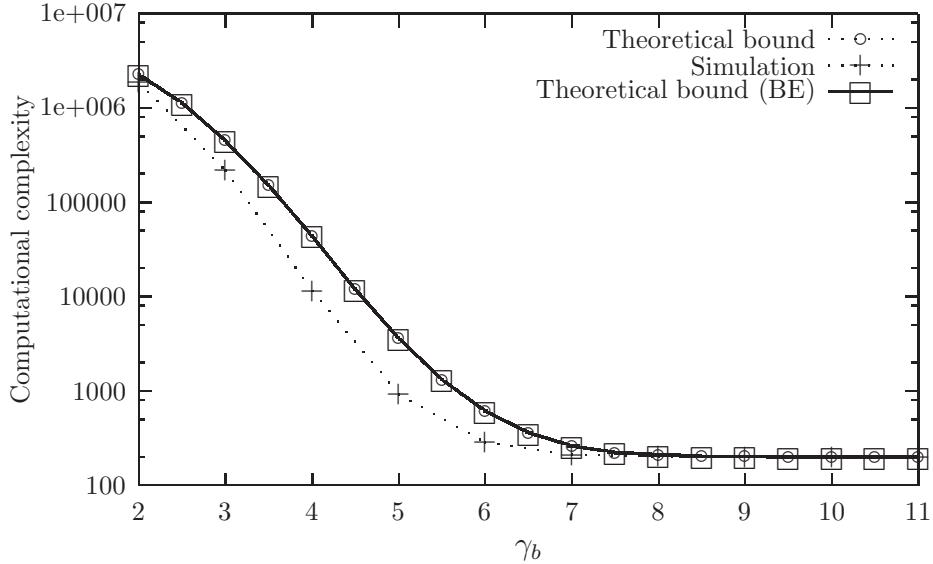


Fig. 8. Average computational complexity of the MLSDA for $(2, 1, 16)$ convolutional code with generators 1632044, 1145734 (octal) and information length $L = 100$.

and Berry-Esseen inequality does not provide much help in decreasing such inaccuracy. As for the MLSDA algorithm, a looser estimate is used to bound the node expanding probability by replacing “summation” by “maximization” as shown in (23). However, the resultant complexity bound is much more accurate simply because the large deviations probability bound is more exact at larger block length.

REFERENCES

- [1] L. E. Aguado and P. G. Farrell, “On hybrid stack decoding algorithms for block codes,” *IEEE Trans. Inform. Theory*, vol. 44, no. 1, pp. 398–409, January 1998.
- [2] Van Beek, “An application of Fourier methods to the problem of sharpening the Berry-Esseen inequality,” *Z. Wahrscheinlichkeitstheorie verw. Gebiete* 23, pp. 183–196, 1972.
- [3] Y. Be’ery and J. Snyders, “Optimal soft decision block decoders based on fast Hadamard transform,” *IEEE Trans. Inform. Theory*, vol. IT-32, no. 3, pp. 355–364, May 1986.
- [4] P. R. Chevillat and D. J. Costello, Jr., “An analysis of sequential decoding for specific time-invariant convolutional codes,” *IEEE Trans. Inform. Theory*, vol. IT-24, no. 4, pp. 443–451, July 1978.
- [5] L. Ekroot and S. Dolinar, “A* decoding of block codes,” *IEEE Trans. on Communications*, vol. 44, no. 9, pp. 1052–1056, September 1996.
- [6] W. Feller, *An Introduction to Probability Theory and its Applications*. volume II, 2nd edition, New York: John Wiley and Sons, 1970.

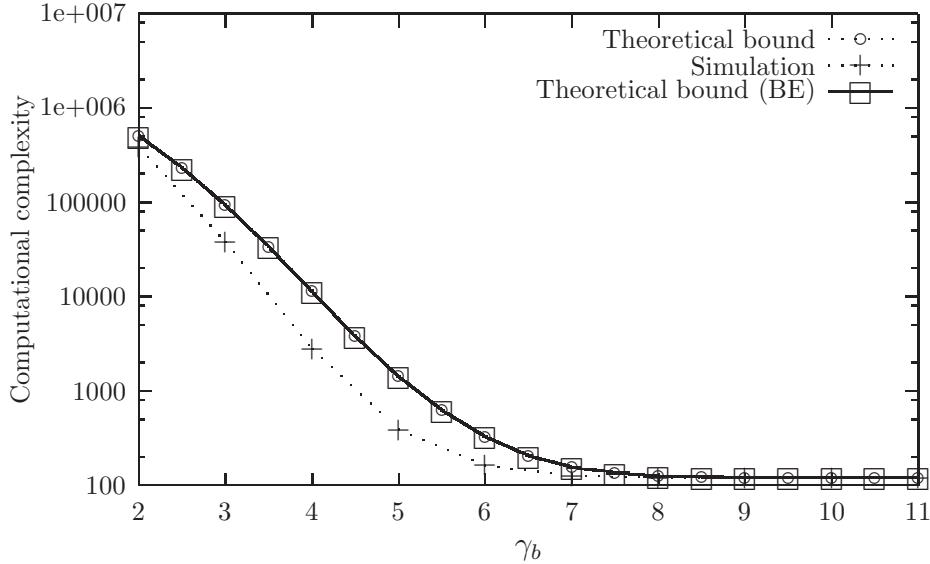


Fig. 9. Average computational complexity of the MLSDA for $(2, 1, 16)$ convolutional code with generators 1632044, 1145734 (octal) and information length $L = 60$.

- [7] G. D. Forney, Jr., “Convolutional codes III: sequential decoding,” *Information and Control*, vol. 25, pp. 267–269, July 1974.
- [8] R. G. Gallager, *Information Theory and Reliable Communication*. New York/John Wiley & Sons, 1968.
- [9] D. Gazelle and J. Snyders, “Reliability-based code-search algorithm for maximum-likelihood decoding of block codes,” *IEEE Trans. Inform. Theory*, vol. 43, no. 1, pp. 239–249, January 1997.
- [10] D. Haccoun, “A branching process analysis of the average number of computations of the stack algorithm,” *IEEE Trans. Inform. Theory*, vol. IT-30, no. 3, pp. 497–508, May 1984.
- [11] Y. S. Han, “A new treatment of priority-first search maximum-likelihood soft-decision decoding of linear block codes,” *IEEE Trans. Inform. Theory*, vol. 44, no. 7, pp. 3091–3096, November 1998.
- [12] Y. S. Han and P.-N. Chen, “Sequential decoding of convolutional codes,” *The Wiley Encyclopedia of Telecommunications*, edited J. Proakis, John Wiley and Sons, Inc., 2002.
- [13] Y. S. Han, P.-N. Chen and H.-B. Wu, “A Maximum-likelihood soft-decision sequential decoding algorithms for binary convolutional codes,” *IEEE Trans. on Communications*, vol. 50, no. 2, pp. 173–178, February 2002.
- [14] Y. S. Han, C. R. P. Hartmann and C.-C. Chen, “Efficient priority-first search maximum-likelihood soft-decision decoding of linear block codes,” *IEEE Trans. Inform. Theory*, vol. 39, no. 5, pp. 1514–1523, September 1993.
- [15] Y. S. Han, C. R. P. Hartmann and K. G. Mehrotra, “Decoding linear block codes using a priority-first search: performance analysis and suboptimal version,” *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 1233–1246, May 1998.
- [16] I. M. Jacobs and E. R. Berlekamp, “A lower bound to the distribution of computation for sequential decoding,” *IEEE Trans. Inform. Theory*, vol. IT-13, no. 2, pp. 167–174, April 1967.
- [17] F. Jelinek, “A fast sequential decoding algorithm using a stack,” *IBM J. Res. and Dev.*, pp. 675–685, November 1969.
- [18] F. Jelinek, “An upper bound on moments of sequential decoding effort,” *IEEE Trans. Inform. Theory*, vol. IT-15, no. 1,

pp. 140–149, January 1969.

[19] R. Johannesson, “On the distribution of computation for sequential decoding using the stack algorithm,” *IEEE Trans. Inform. Theory*, vol. IT-25, no. 3, pp. 323–332, May 1979.

[20] R. Johannesson and K. Sh. Zigangirov, “On the distribution of the number of computations in any finite number of subtrees for the stack algorithm,” *IEEE Trans. Inform. Theory*, vol. IT-31, no. 1, pp. 100–102, January 1985.

[21] F. Kschischang and V. Sorokine, “A sequential decoder for linear block codes with a variable bias term metric,” *IEEE Trans. Inform. Theory*, vol. 44, no. 1, pp. 410–411, January 1998.

[22] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1983.

[23] J. E. Savage, “Sequential decoding-the computation problem,” *Bell Sys. Tech. J.*, vol. 45, pp. 149–175, January 1966.

[24] Senatov, Vladimir V., *Normal Approximation: New Results, Methods, and Problems*, Utrecht, The Netherlands, 1998.

[25] I. S. Shiganov, “Refinement of the upper bound of the constant in the central limit theorem,” *Journal of Soviet Mathematics*, pp. 2545–2550, 1986.

[26] C.-C. Shih, C. R. Wulff, C. R. P. Hartmann and C. K. Mohan, “Efficient heuristic search algorithms for soft-decision decoding of linear block codes,” *IEEE Trans. Inform. Theory*, vol. 44, no. 7, pp. 3023–3038, November 1998.

[27] A. Vardy and Y. Be’ery, “Bit-level soft decision decoding of Reed-Solomon codes,” *IEEE Trans. on Communications*, vol. 39, no. 3, pp. 440–445, March 1991.

[28] A. Vardy and Y. Be’ery, “More efficient soft-decision decoding of the Golay codes,” *IEEE Trans. Inform. Theory*, vol. 37, no. 2, pp. 667–672, May 1991.

[29] K. Zigangirov, “Some sequential decoding procedures,” *Probl. Peredachi Inf.*, pp. 13–25, 1966.

APPENDIX

Proposition 1: For a fixed non-negative integer k , the probability mass of

$$\Pr \{r_1 + \dots + r_d + \min(w_1, 0) + \dots + \min(w_k, 0) \leq 0\}$$

is a decreasing function for non-negative integer d , provided that $r_1, r_2, \dots, r_d, w_1, w_2, \dots, w_k$ are i.i.d. with a Gaussian marginal distribution of positive mean μ and variance σ^2 .

Proof: Assume without loss of generality that $\sigma^2 = 1$. Also, assume $k \geq 1$ since the proposition is trivially valid for $k = 0$.

Let $\Omega_d \triangleq r_1 + \dots + r_d$. Denote the probability density function of w_1 by $f(\cdot)$. Then putting

$\nu \triangleq \Pr\{w_j = 0\}$ yields

$$\begin{aligned}
& \Pr\{\Omega_d + w_1 + w_2 + \cdots + w_k \leq 0\} \\
&= \sum_{j=0}^k \Pr\{\text{exactly } (k-j) \text{ zeros in } (w_1, w_2, \dots, w_k)\} \\
&\quad \Pr\{\Omega_d + w_1 + w_2 + \cdots + w_k \leq 0 \mid \text{exactly } (k-j) \text{ zeros in } (w_1, w_2, \dots, w_k)\} \\
&= \binom{k}{0} \nu^k \Pr\{\Omega_d \leq 0\} + \binom{k}{1} \nu^{k-1} (1-\nu) \int_{-\infty}^0 f(x) \Pr\{\Omega_d \leq -x\} dx \\
&\quad + \binom{k}{2} \nu^{k-2} (1-\nu)^2 \int_{-\infty}^0 \int_{-\infty}^0 f(x_1) f(x_2) \Pr\{\Omega_d \leq -(x_1 + x_2)\} dx_1 dx_2 \\
&\quad + \cdots \\
&\quad + \binom{k}{k} (1-\nu)^k \int_{-\infty}^0 \cdots \int_{-\infty}^0 f(x_1) \cdots f(x_k) \Pr\{\Omega_d \leq -(x_1 + \cdots + x_k)\} dx_1 \cdots dx_k.
\end{aligned}$$

Accordingly, if each of the above $(k+1)$ terms is non-increasing in d , so is their sum. Let

$$\begin{aligned}
q_j(d) &\triangleq \int_{-\infty}^0 \cdots \int_{-\infty}^0 f(x_1) \cdots f(x_j) \Pr\{\Omega_d \leq -(x_1 + \cdots + x_j)\} dx_1 \cdots dx_j \\
&= \int_{-\infty}^0 \cdots \int_{-\infty}^0 f(x_1) \cdots f(x_j) \Phi\left(-\frac{x_1 + \cdots + x_j}{\sqrt{d}} - \sqrt{d}\mu\right) dx_1 \cdots dx_j.
\end{aligned}$$

Then

$$\begin{aligned}
\frac{\partial q_j(d)}{\partial (\sqrt{d})} &= \int_{-\infty}^0 \cdots \int_{-\infty}^0 f(x_1) \cdots f(x_j) \\
&\quad \times \left(\frac{x_1 + \cdots + x_j}{d} - \mu\right) \frac{1}{\sqrt{2\pi}} e^{-(x_1 + \cdots + x_j + d\cdot\mu)^2/(2d)} dx_1 \cdots dx_j \\
&\leq -\frac{\mu}{\sqrt{2\pi}} \int_{-\infty}^0 \cdots \int_{-\infty}^0 f(x_1) \cdots f(x_j) e^{-(x_1 + \cdots + x_j + d\cdot\mu)^2/(2d)} dx_1 \cdots dx_j \quad (24) \\
&< 0,
\end{aligned}$$

where (24) follows from $x_i \leq 0$ (according to the range of integration) for $1 \leq i \leq j$. Consequently, $q_j(d)$ is decreasing in d for d positive and every $1 \leq j \leq k$. The proof is completed by noting that the first term, $\Pr\{\Omega_d \leq 0\} = \Phi(-\sqrt{d}\mu)$, is also decreasing in d . ■